

# Evaluating UAS Autonomy Operations Software In Simulation

Peter A. Jarvis<sup>+</sup>, Robert Harris<sup>+</sup>, and Chad R. Frost

Intelligent Systems Division  
NASA Ames Research Center  
Moffett Field, CA94035  
{ pjarvis, rharris, cfrost }@arc.nasa.gov

## Abstract

We describe a software simulation test bed for evaluating Concepts of Operation (CONOPS) for Unmanned Aircraft Systems (UAS) flying earth science missions. The Mission Operational Concept Evaluation Framework (MOCEF) aids in the rapid evaluation of proposed system automation designs, including intelligent controllers for vehicles, sensor payloads, and decision support systems, on a wide range of missions. Such broad evaluation is prohibitively expensive when limited to physical experiments and real missions.

MOCEF allows evaluation of automation concepts in multiple mission scenarios operating in a wide range of environments. It records mission metric parameters such as the quality of sensor data obtained, flight time, stress on the vehicles, and air traffic control rules infringed or invoked. This information can be fed into specific mission metric formulas to rate performance and into the Google Earth tool for visualization.

**Keywords:** UAS, Simulator, Operations, Automation.

## Introduction

We describe a software simulation test bed for evaluating Concept of Operations (CONOPS) for Unmanned Aircraft Systems (UAS) flying earth science missions. Our Mission Operational Concept Evaluation Framework (MOCEF) aids in the rapid evaluations of proposed system automation designs, including intelligent controllers for vehicles, sensor payloads, and decision support systems, on a wide range of missions. Such broad evaluation is prohibitively expensive when limited to physical experiments and real missions.

We organize this paper as follows. Section 1 describes our motivation for developing MOCEF. Section 2 summarizes the system, identifying its main components and interfaces. Section 3 details the software components that are designed to enable the rapid extension of MOCEF's model set by other users. Section 4 presents our evaluation through the integration with NASA's Apex controller. Section 5 outlines further work.

## Motivation

MOCEF was motivated by the need to evaluate our work on an intelligent controller for a rotorcraft UAS working

on surveillance missions [3]. This controller takes requests to observe a set of locations together, specification of the types of observations to be made at each location, and the value of the information. It then generates a flight plan composed of a set of waypoints for the UAS to fly, together with sensor and communication commands. Our controller remains active during the flight, constantly updating the plan to react to changes in the environment.

When evaluating our controller it became obvious that it is impractical and prohibitively expensive to explore a large range of mission possibilities with test flights. The space of missions, aircraft types, sensor configurations, and weather options is too large.

While there is a wide range of simulators available for testing the flight control software of both manned and UAS, we could not find a suitable simulator for our mission-level controller. We developed MOCEF in part to fill this essential gap in the toolset required in researching intelligent UAS controllers and their application to a variety of operational concepts.

Additional insight came from our experiences with long-endurance UAS missions to observe and monitor wildfires [4]. Decision support systems developed for this and similar Earth science missions proved challenging to evaluate outside of actual flight activities. We elected to develop capabilities in MOCEF to allow us to evaluate the insertion of automation and intelligent systems into a modeled system-of-systems.

A simplified proof-of-concept implementation has been developed and tested, and is described herein.

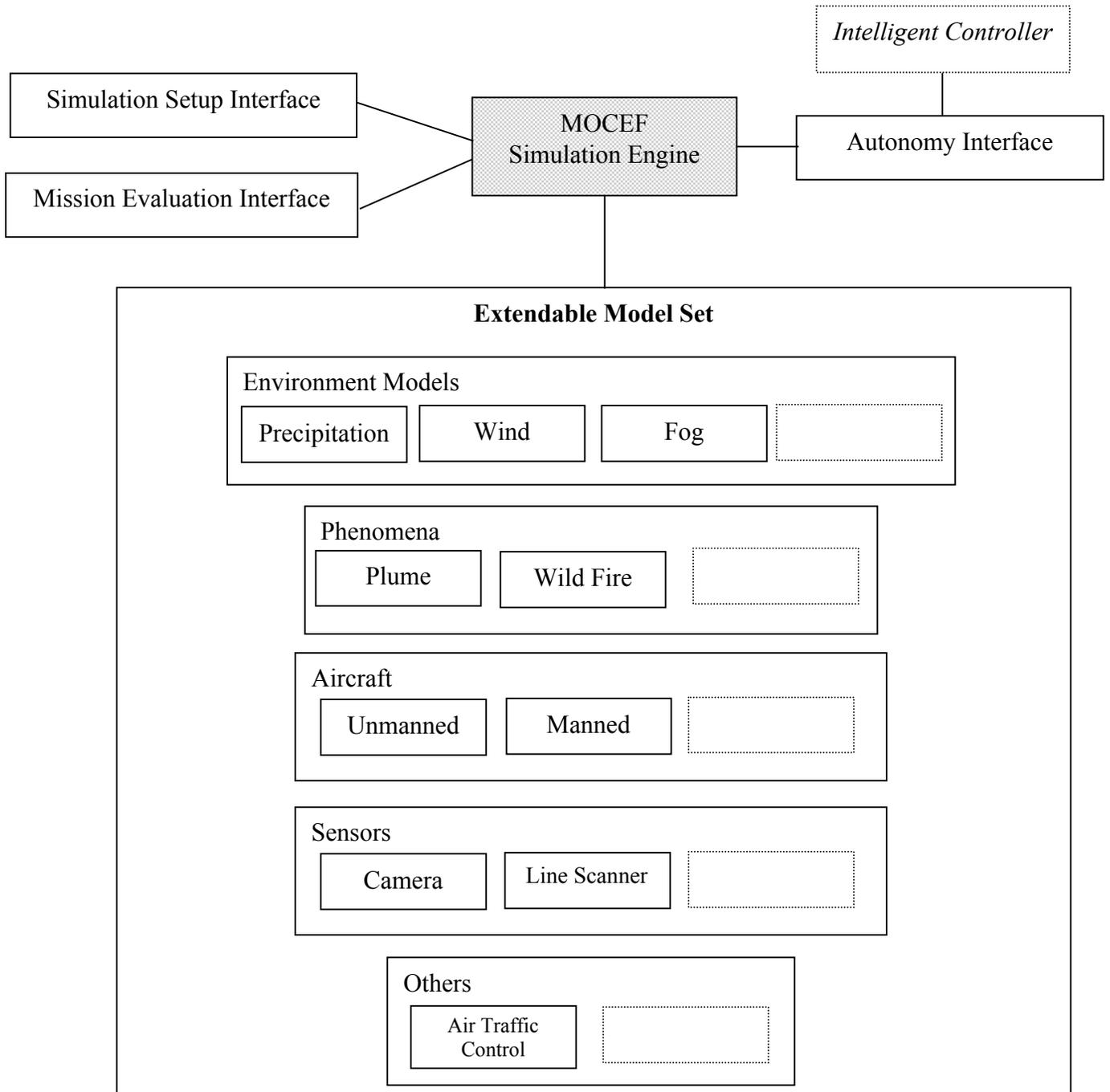
## Architecture

MOCEF is an open extendable simulation framework for evaluating intelligent controllers and other mission-enhancing automation. While we provide a set of default models that are sufficient to evaluate our current UAS work, MOCEF is designed to allow other users to add new models or replace existing models with higher fidelity counterparts. The software prerequisites for MOCEF are low. We require only that a Java virtual machine and optionally the free and cross-platform Google Earth product be installed.

Figure 1 shows the main components of the MOCEF systems. The *MOCEF Simulation Engine* is responsible for orchestrating the simulation. Each model in the system must conform to a common simulation engine interface to receive notifications of when simulated time has advanced

---

<sup>+</sup> PerotSystems Government Services.



**Figure 1: MOCEF Architecture**

and other world state information relayed through the engine.

Stepping through the models first, the *Environment Model* set simulates the physical environment in which the

UAS is operating. The *Precipitation* model defines the areas in the environment where rain is falling. The model includes a path for the rainstorm to take as the simulation progresses and allows the storm to change size. *Wind* is

defined as a direction and speed that makes predefined changes during the simulation. *Fog* is modeled in a similar way to precipitation and other natural phenomena can be readily added to MOCEF. The group of *Phenomena* models represents the entities that our UAS is interested in surveying. Each phenomenon has a prescribed path it will follow during the simulation together with a changing set of physical extents. This allows, for example, a wildfire to move through the simulation growing in size. We also allow a phenomenon's course to be controlled by the wind model by moving from its origin in the direction and at the speed of the current simulated wind.

*Aircraft* follow a flight plan specified as a set of waypoints and speeds to fly between waypoints. In our simulations our autonomy controller constantly adjusts the flight plan for each unmanned aircraft. The simulated manned aircraft follow prescribed flight plans and correspond to the general and commercial aviation aircraft in the environment that our UAS must not interfere with. The *Sensor* models simulate the sensors attached to our UAS. Each sensor executes a sensor plan that is made up of sensing actions and trigger criteria such as aircraft location or simulated time for when those actions should begin and end. The intelligent controller modifies these plans during a simulation in flight with changes in the location of phenomena and the flight plan of the UAS.

The *Air Traffic* model monitors the airspace for possible problems and issues route changes to both the UAS and manned aircraft in our system. It also records violations in separation distance or excursions from the agreed area of operations for use in evaluation of the system.

The *Simulation Setup Interface* allows the user to specify a simulation world setup including initial flight plans for the UAS. Simulation setups are specified in XML syntax and can be authored in any text editor or XML editor.

The *Mission Evaluation Interface* allows the user to observe a simulation and identify what occurred. We provide a highly customizable logging system that can be configured, for example, to log only camera sensing actions and UAS course changes. We also output a trace in Google Earth's KML format for visualization in that tool. Figure 2 shows an example of this output. This facility provides the raw data for input to a mission evaluation tool.

## Detailed Design

We designed MOCEF to support other users in readily extending the model set to meet their simulation requirements. A key component of this strategy was identifying common software functions that could be reused across models. We detail each in this section.

We provide an abstract *route* interface and two instantiations that conform to it. *Prescribed routes* contain a sequence of waypoints, together with the speeds to fly between each pair. *Wind based routes* follow the direction and speed of the wind from a starting point. Many of the

models in our system make use of routes. Aircraft use prescribed routes and phenomena and weather systems tend to use wind-based routes. As both conform to the same interface it is possible to have wind-routed aircraft or prescribed routes for weather items. This general capability can be readily reused in new models.

The concept of an *event trigger* is used throughout our models. A trigger is defined as the pairing of a condition with an action to take when the condition is satisfied. We provide two types of triggers. *Time-based* triggers are activated when the simulation clock reaches a prescribed time. *Position-based* triggers are activated when a specific entity reaches a location in the simulated environment. These triggers can be set to activate only once or each time an entity reaches the specified location. Triggers are used throughout our models; for example, sensors make heavy use of them for specifying when a sensor will be turned on or off. Triggers are also used within phenomena to specify when the entity will grow or shrink in size. The Wind model uses time-based triggers to specify when wind speed and direction will change during the simulation.

We define a *shape* interface for describing the physical extent of entities within the simulation. By allowing entities to be composed of multiple shapes we can model smoke plumes and other three dimensional phenomena. Each shape can be combined with a shape-change plan made up of a set of triggers. These allow the shape of phenomena to change as simulation time moves forward or when the phenomenon reaches a specified point.

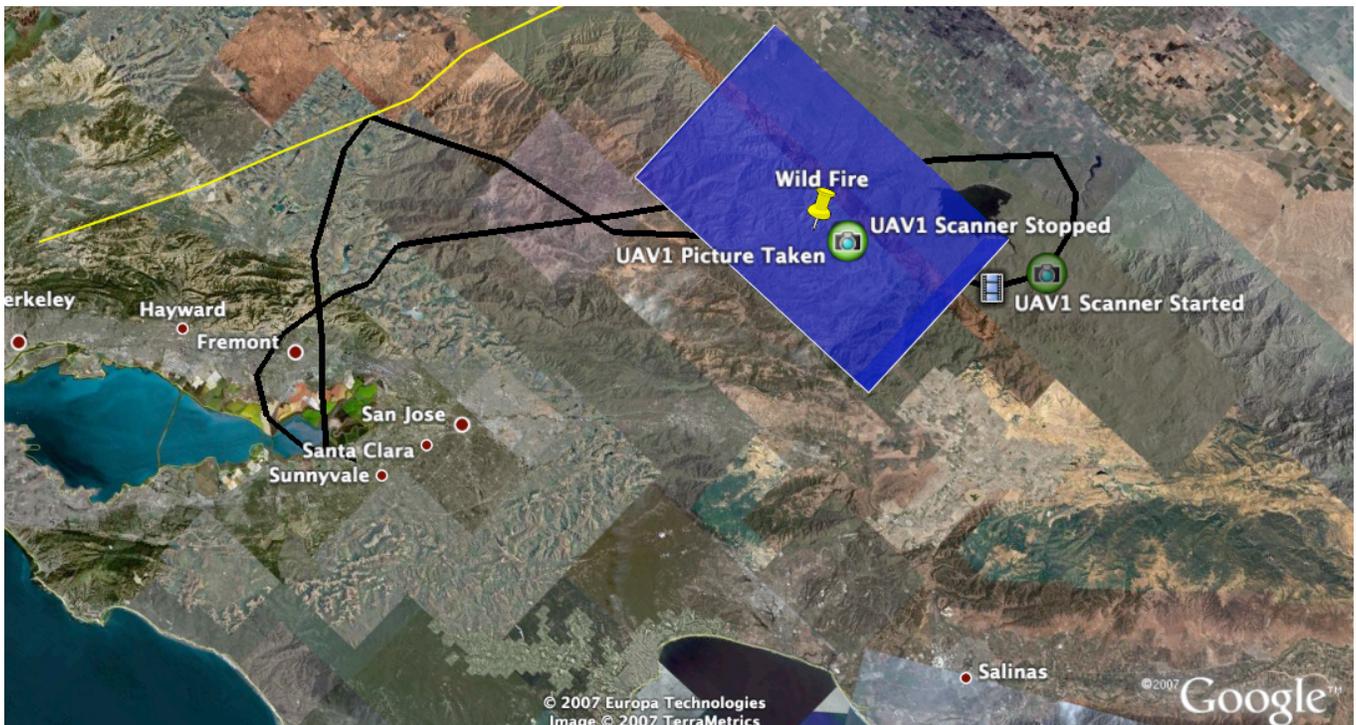
These general concepts are used throughout our MOCEF models and we hope that they will provide useful tools for adding new models to the systems.

## Evaluation

We evaluated MOCEF by running a set of simulated earth science missions controlled by our own Apex-based intelligent UAS controller. Figure 2 shows the Google Earth output produced by MOCEF for one of these missions.

The simulated UAS takes off from the Moffett Field Federal Airfield in California (co-located with NASA Ames) and flies into the San Joaquin valley to observe a simulated wildfire. The Google Earth image traces the path flown by the UAS and the physical extent of the wild fire. The UAS clearly flies over the wild fire and takes pictures and activates a ground scanner.

The Apex controller monitors the UAS and modifies its flight plan as the wild fire moves during the simulation, driven by wind. This proof of concept implementation has confirmed that MOCEF is capable of providing a rich enough environment to test our intelligent UAS controller.



**Figure 2: Google Earth Output**

### Further Work

We plan to make MOCEF open source and available to the broad research community. The main goal of this paper is make researchers aware that this resource is available. We hope that others will add to our existing model set with new models or higher fidelity versions of our current models.

We plan to explore general ways for integrating models that use different geospatial coordinate systems (latitude and longitude, UTM, etc).

### References

- [1] S. M. Schoenung, S. S. Wegener, J. Frank, C. R. Frost, M. Freed, and J. J. Totah. Intelligent UAV airborne science missions. In *AIAA Infotech@Aerospace*, number AIAA-2005-6937, Arlington, VA, September 2005. American Institute of Aeronautics and Astronautics.
- [2] P. Gage, V. Hawke, X. Jiang, E. Lau, and J. Nguyen. A simulation environment for coordinated operations of autonomous vehicles in airspace. Report 01.11.2004, ELORET Corporation, 2004.
- [3] M. Freed, P. Bonasso, K. M. Dalal, W. Fitzgerald, C. R. Frost, and R. Harris. An architecture for intelligent management of aerial observation missions. In *AIAA*

*Infotech@Aerospace*, number AIAA-2005-6938, Arlington, VA, Sept. 2005. American Institute of Aeronautics and Astronautics.

[4] M. V. D’Ortenzio, F. Y. Enomoto, and S. L. Johan, “A collaborative decision environment for UAV operations,” in *AIAA Infotech@Aerospace*, no. AIAA-2005-6939, (Arlington, VA), American Institute of Aeronautics and Astronautics, Sept. 2005.